Monitoring Windows Event Logs

By Ashley Norman

Windows Event Logs are a repository of information about events occurring on a Windows machine. Knowing what is contained in them and how to use that information can help protect oneself in all aspects of the cybersecurity cycle. One way companies do this is through a log monitoring policy. Using this policy in combination with information on how hackers cover their tracks, companies can monitor the event logs for suspicious activity by log type and event ID. A simple way to do this is through a python script which has the benefit of scalability as well as overcoming configuration issues.

Windows event logs are a record of events related to the system, security, and applications stored on a Windows Operating System. They can be used to track system and some applications issues as well as forecast future problems. The elements of a Windows Event Log provide information about hardware and software events occurring on the operating system. The event logs are stored in a standard format which include the main elements, such as Log Name, Event date/time, Task Category, Event ID, Source, and Level. The information contained within the logs depends on the category of event and includes System, Application, Security, Setup, and Forwarded events. The event logs can further contain information related to the type of Windows events and Security level.

A good log monitoring policy is a simple but effective way to protect oneself throughout the cybersecurity cycle. It can provide information to set baselines of “normal” activity, help to identify and prevent attacks, as well as perform post breach forensics. Evidence of security breaches have been found in event logs, but they often go unnoticed because of the large volume of data collected. A large corporation may log thousands of events per second and without a log monitoring system in place this can be overwhelming to tackle. A 2012 Verizon Data Breach report found that “even though 85% of breaches took several weeks to be discovered, 84% of victims had evidence of the breach in their event logs.”

Hackers often try to hide their presence and knowing what to look for is important. A common issue on computers is an application error, which may seem harmless. Frequent application errors and crashes could actually be a sign of an attacker covering their tracks from inserting malware or a virus on your computer. A search of the application logs for Event ID 1000 can provide more information. Other application log events to monitor are Event ID 1002, which is generated when an application has trouble loading. Event ID 4719, a system audit policy that was changed could also show malicious behavior in the application logs.

A popular attack is Pass the Hash which allows hackers to gain access to an account without needing to know the password. This was used as recently as April of this year by a Ransomware as a Service platform called Hive. This attack leveraged the hash technique to target a large number of Microsoft’s Exchange Server customers. While there are several mitigation strategies to prevent and detect this, monitoring the event logs is one such way. Event IDs associated with logins can help mitigate this type of attack. Event ID 4624

(successful login), 4625 (failed login), and 4648 (login with special credentials) are logs to use in combination to monitor for this type of attack by looking at suspicious levels.

Developing a python script to monitor these log types by event ID could help in detecting any suspicious activity. Python is an easy to use backend programming language that can be used for scripting. By using python in combination with the pywin32 module you can develop a script to monitor the event logs. The pywin32 module lets you access the features of the Windows Event Logs in python. The benefit of this python script is that it allows for scalability when applying to a large organization and you may need to search for events across hundreds or even thousands of machines. The other benefit is when you have a SIEM that is not configured to consume Windows logs, this script would be a way to access that. This was an issue the MN Air National Guard was experiencing and they recently utilized this script to conduct host based forensics.

In conclusion, Windows Event Logs are a breadth of knowledge about the events occurring on a Windows machine. Frequent monitoring as well as knowing how to parse through the information can be beneficial in protecting oneself against malicious activity. A python script is just one way you can conduct event monitoring as part of your log monitoring policy.

References

- https://www.solarwinds.com/resources/it-glossary/windows-event-log
- https://www.beyondtrust.com/blog/entry/windows-server-events-monitor
- https://www.cyberark.com/resources/threat-research-blog/detecting-pass-the-hash-with-windows-event-viewer
- https://www.crowdstrike.com/cybersecurity-101/pass-the-hash/
- https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/monitoring-active-directory-for-signs-of-compromise
- https://stackoverflow.com/questions/11219213/read-specific-windows-event-log-event
- https://www.coursereport.com/blog/what-is-python-programming#what-is-python